NATURAL LANGUAGE PROCESSING FOR JOINT FIRE OBSERVER TRAINING

Antonio Roque, Kallirroi Georgila, Ron Artstein, Kenji Sagae and David R. Traum*
USC Institute for Creative Technologies
12015 Waterfront Drive
Playa Vista, CA 90094-2536

ABSTRACT

We describe recent research to enhance a training system which interprets Call for Fire (CFF) radio artillery requests. The research explores the feasibility of extending the system to also understand calls for Close Air Support (CAS). This work includes automated analysis of complex language behavior in CAS missions, evaluation of speech recognition performance, and simulation of speech recognition errors.

1. INTRODUCTION

Virtual environments such as the Joint Fires and Effects Trainer System (JFETS) can help provide best-in-class training in Call for Fire (CFF) radio artillery request skills. In the JFETS training environment, we have previously investigated using spoken dialogue systems to automate routine radio dialogues (Roque et al., 2006). In this paper, we describe more recent research on spoken natural language processing to enhance the training environment through increased automation and understanding of more complex dialogues. This work includes improved semi-automated dialogue systems, analysis of complex language in Close Air Support (CAS) missions, evaluation of speech recognition performance, and finally, simulation of speech recognition errors.

Due to the complexity of the training tasks and the rich nature of the JFETS virtual environment, it is neither desirable nor feasible to eliminate human operators from the training system. However, many of the tasks an operator performs are routine and can be automated. The Intelligent Operator Training Assistant (IOTA) is designed to handle many of the routine tasks, freeing the operator to focus only on the "out of the ordinary" situations that occur, and the specific educational needs of the soldier. This has the potential to multiply the human operator's efficiency by enabling a lone operator to singly manage several training sessions with multiple soldiers in parallel. In some cases, when the soldier performs the task within pre-defined parameters, the whole JFETS training session might be handled by the IOTA. In other cases, where the soldier departs from pre-defined parameters, the human operator is able to take over control of the session from the IOTA until the soldier is back within the established parameters. We enable this flexibility by tightly integrating the intelligent aspects of IOTA with the human-controlled aspects of JFETS.

In section 2, we provide a brief description of the current IOTA capabilities, and in the remaining sections we discuss how these capabilities can be extended and improved by using additional technologies for processing natural language. In particular, we conducted research and analysis of dialogue used during CAS missions, towards extending the system's capabilities beyond the CFF missions currently handled. We examined word differences between CAS and CFF missions, as described in section 3.

Next, we noted that differences in vocabulary and dialogue moves are likely to affect IOTA's Speech Recognition component, which translates soldier utterances into text. To study this, we evaluated the performance of several speech recognizers on a corpus of CFF+CAS missions. We also evaluated the same recognizers on a corpus of CFF missions only. This is described in section 4.

We also noted that dialogues in CAS missions often contain less constrained verbal interactions that include conversational sentences with standard English structure, which require more sophisticated machinery for automatic extraction of information and analysis of dialogue acts (even if restricted to a particular domain). CAS dialogues contain both structured utterances (like call-sign identification), which can be handled by existing IOTA technology, as well as more conversational language (such as free-form target descriptions), which is beyond the capability of the current deployed system. In section 5, we propose a Natural Language Processing (NLP) pipeline for analysis of CAS utterances.

Very often, especially when designing a new application there is a shortage of data for training and evaluating the speech recognizer, which makes it very difficult to predict the Word Error Rate (WER) when the system is being used by real users. Ideally we would want to ensure that our system performs well for a variety of WERs. We performed a corpus study to see whether we

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2010		**00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Natural Language Processing for Joint Fire Observer Training	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Southern California,Institute for Creative Technologies,12015 Waterfront Drive,Playa Vista,CA,90094-2536**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the of the 27th Army Science Conference, Orlando, FL November 29 - December 2, 2010. U.S. Government or Federal Rights License

14. ABSTRACT
We describe recent research to enhance a training system which interprets Call for Fire (CFF) radio artillery requests. The research explores the feasibility of extending the system to also understand calls for Close Air Support (CAS). This work includes automated analysis of complex language behavior in CAS missions, evaluation of speech recognition performance, and simulation of speech recognition errors.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**8**	

Table 1: Vocabulary differences

	CAS	CFF
Protocol	cleared, hot, egress, contact, reciprocal, standby, wheel, read back, nine line, tally target, JTAC qualified	adjust, fire, effect, polar, distance, add, drop, message, observer, shot, splash, rounds, complete, target number, my command, immediate suppression
Compass	north, south, east, west etc.	
Munitions	a_g_m (air ground missile), a_tens, bombs, f_sixteens	h_e (high explosive), i_c_m (improved conventional munitions), illumination, w_p (white phosphorous)
Enemies	manpads, r_p_gs, trucks	b_r_d_m, b_t_r, infantry
Call signs	hog, talon, tulsa	thunder, cherry

can reliably simulate speech recognition errors. This is described in section 6.

The results of these studies indicate that technology already exists that can enhance training of forward observers by automating some language understanding tasks, particularly for structured domains such as CFF, however more research is needed to be able to improve processing of more complex, less constrained language use in CAS missions.

2. CURRENT IOTA CAPABILITIES

IOTA has several features to assist the operator:

- When enabled, IOTA automatically updates the CASTrainer interface with the relevant CFF information. For example, if the Soldier provides a grid coordinate, IOTA will extract the relevant digits and insert them into the 'grid' text-area where the operator would have inserted them, and IOTA will also print the Soldier's utterance to the Mission Status text-area. In this way, IOTA keeps an up-to-date record that the operator can use to quickly recover the context of a given training session.
- When enabled and managing a CFF, IOTA will track the information that has been given by the Soldier, and IOTA will fire the mission when it has enough information.
- If it encounters problems while managing an interaction, IOTA will attempt to notify the operator through the text-area.
- IOTA records logs and sound files, which can be analyzed for further information about student behavior.

To achieve this functionality, IOTA consists of the following components. First, an Automated Speech Recognizer component takes the voice signal, and translates this into text (see sections 4 and 6). Next, an Interpreter component determines what the meaning of the text is: whether a warning order is given, or a target location, or some other dialogue move, and if so, what the parameters are (see section 5). A Dialogue Manager component determines whether a voice confirmation is needed, and if so, uses a Text-to-Speech engine to produce it. The Dialogue Manager also determines what kind of command is needed for sending to the JFETS CASTrainer, and produces that if so.

As we will see in the following, IOTA technology shows promise to also handle CAS types of missions with further analysis and development.

3. CORPORA AND VOCABULARY DIFFERENCES

CAS missions have a different protocol and refer to different objects than CFF dialogues, so one would expect that the words used are also different. This section considers the vocabulary differences between the two domains. The analysis is based on two corpora from the JFETS training environment in Ft. Sill, Oklahoma:

- IOTA-2008 was collected from January to July 2008, and contains speech of both the trainee and the operator. The recordings have been manually transcribed, tokenized and tagged with dialogue acts by the system's classifier, and then corrected by hand and separated manually into CAS dialogues (69369 words) and CFF dialogues (24792 words).
- OTM-2009 was collected from August to October 2009, and contains the speech of the trainee only. The recordings have been manually transcribed and separated into CAS dialogues (24497 words) and CFF dialogues (27028 words).

There are substantial differences between the vocabularies of CAS and CFF dialogues. Table 1 shows a few samples of words and bigrams (word pairs) which occur at least 10 times more frequently in one dialogue type.

Protocol words. This is an obvious difference, as the radio protocols differ between the two types of dialogues.

Points of the compass are very frequent in CAS dialogues and almost completely absent in CFF dialogues.

Munitions and platforms are not strictly part of the protocol, but they tend to differ between the two domains.

Scenario features. In principle, both CAS and CFF can be called for the same scenarios, and our corpora contain some joint exercises which mix calls from the two domains. For the most part, however, CAS and CFF exercises use distinct scenarios with different enemies.

Call signs. There is no principled reason for having different call signs, but in our corpora they differ.

The vocabulary differences make it very easy to identify whether a dialogue belongs to the CAS or CFF domain, though there may be some difficulty in precise segmentation of joint exercises.

4. SPEECH RECOGNITION PERFORMANCE EVALUATION

In this section we evaluate the performance of several speech recognizers on a corpus of CFF missions and a corpus of CFF+CAS missions. Since ASR systems are typically tuned to the environment they operate in, performance is affected by many factors, among them: the domain/vocabulary that the recognizer is expected to handle, the acoustic environment in which the recognizer operates, and the speech recognition engine. Additionally, there is often a trade-off between the quality of the speech recognition output and the time it takes to reach that output; real-time conversational systems may be willing to accept a somewhat degraded output in return for lower latencies. When comparing CFF missions with CFF+CAS missions, we attempted to consider as many of these issues as possible.

4.1. Corpora Used

We used two sets of data for this comparison:

- Radiobots - This speech data was collected in 2006 in JFETS at Fort Sill, Oklahoma, with volunteer trainees who performed calls for specific missions (Robinson et al., 2006). This corpus contains only CFF missions.
- IOTA - This speech data was collected in 2008-2009 in JFETS at Fort Sill, and includes both CFF and CAS missions.

All utterances were transcribed manually. We split each data set randomly into training, development, and test sets: development and test sets were each slightly over 10% of the turns for each corpus, with the remainder used for training. The size of the data sets is shown in Table 2.

Table 2: Training data sizes (Words/Turns)

	TRAIN	TEST	DEV
Radiobots	6841/1082	1163/167	1325/190
IOTA	49633/4939	5441/650	6552/608

4.2. Approach

The following recognizers were used:

- Cambridge HTK family: HVite (v3.4.1), HDecode, AVite (v1.6), Julius (v4.1.2)
- CMU Sphinx family: Sphinx 4, Pocket Sphinx (v0.5)

Acoustic models and language models were first trained on the training set (TRAIN). Then the recognizers were tuned on the development set (DEV) and the final result was calculated on the test set (TEST). More details about the training procedure are provided in (Yao et al., 2010).

Our evaluation metrics were word error rate (WER) and recognition speed. WER can be formulated as:

$$WER = \frac{S + D + I}{N} \times 100\,\%$$

where S, D and I are the number of substitutions, deletions and insertions respectively, and N is the length of the target string (i.e. the string of words that the Soldier uttered). Speed is measured by whether the recognition was real-time or not. A real-time recognizer can finish recognizing a segment of speech in a time interval no greater than the length of the speech.

4.3. Results

Tables 3 and 4 show the performance of the various recognizers on the different data sets. For each recognizer, the left column shows the best WER achieved on DEV after tuning the parameters; the right column shows the performance of the same parameter settings on TEST.

Table 3: Non–real time speech recognition results

	HVite		HDecode		Sphinx4	
	Dev	Test	Dev	Test	Dev	Test
Radiobots	10	15	11	12	-	-
IOTA	66	57	49	39	76	-

Table 4: Real time speech recognition results

	Julius		AVite		PktSphx	
	Dev	Test	Dev	Test	Dev	Test
Radiobots	17	14	12	-	7	10
IOTA	61	42	-	-	55	47

4.4. Conclusion

Two observations from the tables are notable. First, no one recognizer dominates on all data sets. Second, conversational speech recognition is still a challenging task with high WERs for IOTA, which used CAS as well as CFF dialogues. For more experiments and results see (Yao et al., 2010).

5. SYNTACTIC AND SEMANTIC ANALYSIS OF CAS DIALOGUES

While dialogues in CFF missions tend to follow a somewhat controlled structure, where information can be extracted successfully using an approach that identifies patterns based on the linear sequence of words (known as sequence labeling techniques), as shown by Roque et al. (2006) in the IOTA system, dialogues in CAS missions often contain less constrained verbal interactions that include conversational sentences with standard English structure. This results in a larger vocabulary and generally richer syntactic and semantic structure in the language used in CAS, which require more sophisticated machinery for automatic extraction of information and analysis of dialogue acts. While much of the IOTA technology is applicable to a portion of these utterances, further development that accounts for richer language usage would provide additional language understanding capabilities to the system, opening possibilities for extensions that allow IOTA to handle CAS missions.

Consider, for example, the following two utterances, taken from manually transcribed CAS dialogues:

1. target location two seven five degrees
2. once you get to that village you see a uh almost looks like a martini glass at the south end of that lake

The information contained in the first utterance can be identified with a simple template, or with a sequence labeling technique similar to the one used in IOTA for automatic interpretation of utterances in CFF dialogues (Roque et al., 2006). Utterances such as this, from which all or most useful information can be extracted without structural syntactic or semantic analysis, occur throughout the corpus used in our analysis (about 7,000 CAS utterances), but amount to less than 10% of all utterances in that corpus. The current approach used in IOTA would also be suitable for other utterances that do contain meaningful, but simple, standard English syntactic structure (e.g. *charlie four two this is goblin*).

The second utterance contains information about an event encoded in a more complex syntactic structure, which includes, for example, a temporal clause (the phrase *once you get to that village* refers to the time of the event), and words with meanings that cannot be determined in isolation (the word *get* in this utterance has a meaning similar to "reach a destination," but this is only apparent when the rest of the utterance is taken into account). Utterances that contain this type of general conversational language amount to more than 70% of our corpus, and would require more than special-purpose pattern matching rules or a sequence labeling approach for accurate and comprehensive extraction of information or fine-grained classification of dialogue moves and parameters. The analysis in this section focuses on such utterances.

5.1. An Illustrated Example

To illustrate the type of information that can be identified using NLP approaches, we show the information we hope to obtain from a specific CAS utterance using a syntactic parser and a semantic-role analyzer in Figures 1 and 2, respectively. Note that all aspects of analyses would be obtained completely automatically from utterance 2 above.

While the syntactic analysis of the utterance (Figure 1) does not reflect directly the meaning intended by the speaker, it does provide useful information that can be used in the identification of dialogue moves and parameters associated with this utterance. For example, knowing that this utterance was produced by the operator, it can be trivially inferred that *you* refers to the soldier, who is the subject of an action (*see*). Syntactic analysis can also provide information about spans of words that may form meaningful units. This type of analysis is also used as the input for the semantic role analyzer, which produces the output shown in Figure 2. In this semantic role analysis, which contains more of the meaning in the utterance, we see that a proper meaning was assigned to the predicates *see*, *get*, and *look*. This is not a trivial task, as these words may have very different meanings in different contexts. This analysis shows, among other things, that the utterance is about a *viewing* event, where the viewer is the soldier (*you*), that occurs when the soldier reaches the village. This type of analysis is more challenging to perform accurately than the purely syntactic analysis.

Utterance: *Once you get to that village you see a uh almost looks like a martini glass at the south end of the lake.*

Syntactic information:

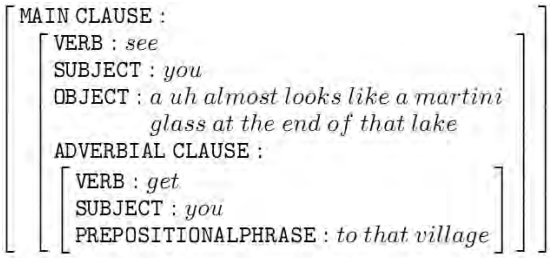

Figure 1: Syntactic information obtained from a CAS utterance using a syntactic parser

Semantic roles:

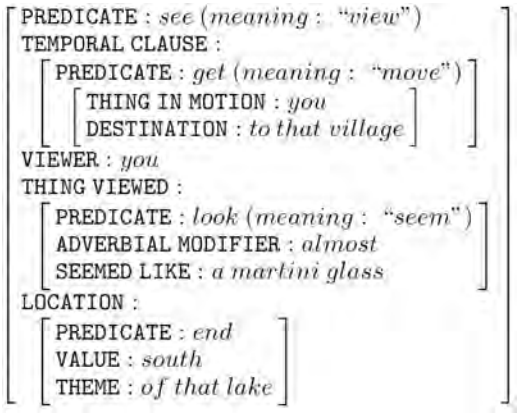

Figure 2: Semantic roles corresponding to the utterance and syntactic information of Figure 1

5.2. A Natural Language Processing Pipeline for Analysis

The first step towards the application of NLP techniques to CAS utterances is identification of the specific utterances for which these techniques are expected to be effective. Then a sequence of NLP modules perform different levels of analysis at the word-level (part-of-speech tagging), structural phrase level (syntactic analysis), utterance level (utterance segmentation), and finally semantic level (semantic role analysis). We outline the challenges and techniques involved in each of these steps below. We generally base our NLP methodology on data-driven methods, which learn desired behavior from a set of manually annotated examples. Data-driven NLP approaches have been shown to offer high levels of accuracy and robustness to noisy input. It is important to keep in mind that the work discussed here relies heavily on the accuracy of the transcriptions used as input for our NLP pipeline. At the current level of speech recognition accuracy for CAS utterances in IOTA (described in the previous section), performance of NLP technology would be severely degraded. Therefore, successful application of the work discussed below (based on manual transcriptions) in a run-time system depends on improved speech recognition for CAS utterances. Alternatively, NLP could be used for off-line analysis of manually transcribed data.

Identification of candidate utterances to be analyzed using NLP techniques is a fairly straightforward task that can be accomplished using existing utterance classification approaches (e.g. Sagae et al., 2009), where machine learning techniques are used to determine utterance types. Even a simple filter that checks whether more than one third of the words in each utterance is composed of digits, month names or spoken alphabet words has over 90% accuracy (based on a random sample of 300 utterances from our corpus of CAS dialogues) in selecting utterances from which NLP modules can recover useful information.

Once utterances are selected for syntactic and semantic analysis, the next step is to identify word classes, such as nouns, verbs, adjectives, and adverbs. This task is commonly referred to as part-of-speech (POS) tagging. Current approaches for POS tagging use statistical models based on hundreds of thousands of words that have been manually tagged with correct categories, and can achieve accuracy levels above 97% on news articles in English (Tsuruoka and Tsujii, 2005). To process the more spontaneous and conversational utterances in CAS dialogues, we trained the POS tagger described by Tsuruoka and Tsujii (2005) using the manually annotated Switchboard section of the Penn Treebank (Marcus et al., 1993; Bies et al., 2005), which contains part-of-speech and syntactic structure annotation for roughly one million words of transcribed telephone conversations. As should be expected, the resulting tagger makes incorrect POS tag assignments when faced with language usage missing from its training data, such as in call signs and other domain-specific words and phrases, such as “roger” and “good burn.” These problems would be solved with CAS-specific training data.

To determine the syntactic structure of CAS utterances (Figure 1), we use dependency parsing, which is a syntactic analysis approach well-suited for conversational language. Application of commonly used off-the-shelf parsers built for analyzing written text produced syntactic structures that contained a large number of crucial errors in the analysis of CAS utterances.

These errors were caused in large part by disfluencies and domain-specific vocabulary and structure. As with POS tagging, we adapted an existing dependency parser (Sagae and Tsujii, 2007) for conversational language using the Switchboard portion of the Penn Treebank. The accuracy of the resulting parser, measured as the percentage of correct word-to-word relationships in the parser's output (the standard measure for dependency parsing accuracy in the NLP literature), in a small pilot evaluation using a set of 100 utterances was 86%, suggesting that this is a promising approach, and that accurate analysis of CAS utterances is feasible. This result also indicates that the accuracy of the POS tagging approach based on Switchboard data is sufficient to support syntactic analysis.

The output of the syntactic parser can be used in other modules that could perform dialogue act prediction or utterance segmentation, but it does not include a direct representation of the meaning of the utterance. In cases where a more semantically-oriented analysis is needed, another layer of processing called Semantic Role Labeling (SRL) can be applied. SRL can determine the intended usage of verbs (Figure 2), as well as label the participants in events with their appropriate roles. However, SRL technology is not as well developed as syntactic parsing, and the level of performance that can be expected in language that differs from news text is largely unknown. It is possible that an SRL system that uses existing resources (training material and dictionaries) with minor modifications could achieve high levels of accuracy, given that the language domain is sufficiently narrow, and that the accuracy of the adapted parsing module is relatively high, which is an important factor for SRL accuracy. We have integrated such an SRL module in our NLP pipeline, and although an evaluation is necessary to determine the suitability of this technology to IOTA, initial results do not rule it out. For example, the sample syntactic and semantic-role analyses presented in our illustrated examples were in fact generated fully automatically with the pipeline we have described.

5.3. Conclusion

We have found that current data-driven NLP technology can be successfully adapted and applied to IOTA for the analysis of CAS utterances. Use of these techniques in a run-time system would also require improvements in speech recognition accuracy for these utterances. Even in the absence of improved speech recognition, NLP could still be useful in off-line analysis of manually transcribed dialogues.

6. SPEECH RECOGNITION ERROR SIMULATION

As we saw above, speech recognition is a very hard problem for the IOTA data set (CFF+CAS missions). Very often, especially when designing a new application, there is a shortage of data for training and evaluating the speech recognizer, which makes it very difficult to predict the WER that the system will have interacting with real users. Ideally we would want to ensure that our system (in particular, the Interpreter component and the Dialogue Manager) performs as well as possible for a variety of WERs.

We performed a study using the IOTA data set to see whether we can reliably simulate speech recognition errors. Our goal is to test two hypotheses. Our first hypothesis is that it is possible to train models for simulating speech recognition errors, and by adjusting some parameters generate different WERs. Our second hypothesis is that it is possible to generate simulated errors with a distribution similar to the distribution of errors observed with a real speech recognizer.

Given a source utterance, our goal is to generate a "scrambled" target utterance so that, when comparing the source and the target utterances, the resulting WER is similar to the WERs we observe with a real speech recognizer. Consider the example below where the word "direction" is scrambled and becomes "direction six" resulting in a WER of 20%.

Source utterance:	direction	two	zero	four	five
Target utterance:	direction six	two	zero	four	five

6.1. Approach

The problem of simulating speech recognition errors has attracted much attention in the literature, especially as an integral part of a simulated user (Georgila et al., 2006). The idea of using phonetic confusions for speech recognition error simulation has been explored by many groups including (Fosler-Lussier et al., 2002; Pietquin, 2004). The above approaches produce promising results but often require a large amount of training data. A computationally less expensive approach is to measure the confusability of each word in the corpus by counting how many other words it is confused with. However, this approach does not take into account the context of each word.

Here we use an approach that is computationally inexpensive and at the same time avoids the disadvantages of considering words in isolation. Our approach is similar

to the one presented in (Schatzmann et al., 2007) with a few modifications, mainly implementation issues. Following (Schatzmann et al., 2007), at the word level, speech recognition error simulations can be viewed as translations of a source utterance w to a scrambled utterance u. The source utterance can be described as a sequence of S words, $w_{1,S}$, or a sequence of N fragments, $f_{1,N}$, where each fragment is a group of contiguous words in w. In the same way, the target utterance u can be viewed as a sequence of T words, $u_{1,T}$, or a sequence of N confused fragments, $f_{1,N}$. Note that while S and T may be different we can assume that the number of N "clean" source fragments can match the number of "scrambled" target fragments. This is because each fragment can have 0 or more words. An example is given in Figure 3.

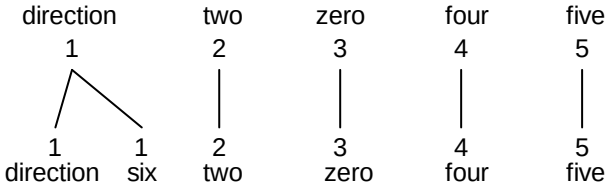

Figure 3: A sample source and target alignment

During training we use pairs of reference transcriptions and speech recognition outputs and align them using a Levenshtein distance matrix such that the transformation of the reference transcription into the speech recognition output is done with the minimal number of insertions, deletions, and substitutions. The result is a lookup table of all fragments occurring in the training transcriptions, together with the possible scrambled fragments and the frequency of each mapping. Example mappings for the fragment "back to" can be seen below. "Back to" can be scrambled as "back" with a probability 80% and as "to" with a probability 20%.

back to	back	→	8
back to	to	→	2

We build two language models (back-off 3-grams), one based on the speech recognition outputs (language model L1) and one on the fragment scramblings and in particular only the source fragments (in the example above "back to"), which we call language model L2.

During testing the algorithm has two tasks. First to split the source utterance into fragments and then apply the most appropriate scramblings of these fragments so that the desired WERs are accomplished (first hypothesis), and the distribution of WERs observed with real speech recognition is preserved (second hypothesis).

The algorithm works as follows: Consider the source word sequence $w_1,w_2,...,w_M$. The word w_1 is necessarily assigned to fragment f_1. Let p_1 be the probability of seeing w_{i-1} alone in a fragment (based on the language model L2) and p_2 be the probability of seeing w_i follow w_{i-1} in the fragment (again based on the language model L2). If $p_2 > p_1$ then w_{i-1} and w_i will be part of the same fragment and we can continue in the same way to see whether w_{i+1} will be part of the same fragment or start a new fragment. If, on the other hand, $p_1 > p_2$ then w_i will start a new fragment. For more details see (Schatzmann et al., 2007).

In the following, our approach deviates from the method of Schatzmann et al. (2007). Now that we have decided on the fragments, for each fragment we apply all possible scramblings above a threshold P. The next step is to use the Viterbi algorithm and select the combination of scramblings along the whole sentence that will lead to the highest overall probability. Here we use the language model L1.

6.2. Evaluation

To test our hypotheses we used the IOTA data set (the same as used for the speech recognition evaluation experiments in section 4). The data set used for the reference transcriptions in training (TRAINsim) is the same as the one used for testing in the speech recognition evaluation section (TEST) since it is the most appropriate set for giving us correct distributions of real WERs. For speech recognition outputs we used the output of Julius on TEST, which produced the best result we got on IOTA with real-time speech recognizers. For testing on unseen data we used the data set TESTsim (equivalent to TRAIN).

In the following table we can see the simulated WERs generated by applying the algorithm on TESTsim for different thresholds P. As we can see low thresholds P lead to high WERs. Having a low P means that we allow for scramblings that did not appear frequently in the training data. With a high P, the less frequent confusions will be ignored, which of course will contribute to a lower WER. Note that with a threshold P=0.001 we can simulate the WER of Julius quite accurately. The results below satisfy our first hypothesis. It is therefore possible to generate different WERs by adjusting some parameters.

Table 5: Simulated WERs for various thresholds

Threshold P	0.001	0.050	0.100	0.200	0.400
WER(%)	44.6	29.0	17.0	9.7	3.8

In Table 6, we can see the mean Word Error Rate (mWER) and its standard deviation (sdWER) observed

with a real recognizer (Julius) on TRAINsim (the same as TEST for the speech recognition evaluation in section 4), and the mWER and sdWER observed on the sentences generated by the error simulation algorithm on both TRAINsim and TESTsim. mWER differs from WER, as presented in Table 5 in that WER is calculated over the whole corpus, while mWER is the average of WERs for each utterance, and thus mWER gives greater weight to words in short utterances than words in long utterances, while WER gives the same weight to all words in the corpus.

Table 6: Mean and standard deviation for real and simulated WERs

	TRAINsim		TESTsim	
	mWER	sdWER	mWER	sdWER
Julius	34	29	-	-
Simulator	33	22	32	21

Our result shows that it is possible using unseen data (TESTsim) to generate errors with a distribution very similar to the distribution of errors observed with a real speech recognizer. Julius produced a distribution of WERs with mean 34 and standard deviation 29. Our algorithm produced a distribution of WERs with mean 32 and standard deviation 21.

6.3. Conclusion

Using the IOTA data set, we found that it is possible to train models for simulating speech recognition errors, and by adjusting some parameters generate a variety of WERs. We also showed that it is possible to generate simulated errors with a distribution similar to the distribution of errors observed with a real speech recognizer.

7. OVERALL CONCLUSIONS

We have examined CAS dialogues in a number of ways, focusing on differences from CFF missions in terms of vocabulary, dialogue act, and speech recognition performance. Although there are recognizable differences between CFF and CAS missions, IOTA technology shows promise to handle CAS types of missions with further analysis and development.

In our future work, we hope to do more annotations in order to develop and test domain-specific versions of the components presented in section 5. Furthermore, our speech recognition error simulator will enable us to experiment with different WERs, and thus see which range of WERs the techniques of section 5 and generally the full IOTA system will work for, so we will be ready as improvements to speech recognition are made to leverage the most appropriate technologies.

ACKNOWLEDGMENTS

This work has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

REFERENCES

A. Bies, J. Mott, and C. Warner. 2005. Addendum to the Switchboard Treebank Guidelines. *Linguistic Data Consortium.*

E. Fosler-Lussier, I. Amdal, and H.-K. J. Kuo. 2002. On the Road to Improved Lexical Confusability Metrics. In *Proceedings of PMLA.*

K. Georgila, J. Henderson, and O. Lemon. 2006. User Simulation for Spoken Dialogue Systems: Learning and Evaluation. In *Proceedings of Interspeech.*

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313-330.

O. Pietquin. 2004. A Framework for Unsupervised Learning of Dialogue Strategies. *Ph.D. Thesis, Polytech de Mons*.

S. Robinson, A. Roque, A. Vaswani, C. Hernandez, B. Millspaugh, and D. Traum. 2006. Evaluation of a Spoken Dialogue System for Virtual Reality Call for Fire Training. In *Proceedings of 25th Army Science Conference.*

A. Roque, A. Leuski, V. Rangarajan, S. Robinson, A. Vaswani, S. Narayanan, D. Traum. 2006. Radiobot-CFF: A Spoken Dialogue System for Military Training. In *Proceedings of Interspeech.*

K. Sagae, G. Christian, D. DeVault, and D. Traum. 2009. Towards Natural Language Understanding of Partial Speech Recognition Results in Dialogue Systems. In *Short Paper Proceedings of the NAACL-HLT.*

K. Sagae, and J. Tsujii, J. 2007. Dependency Parsing and Domain Adaptation with LR Models and Parser Ensembles. In *Proceedings of EMNLP-CoNLL*.

J. Schatzmann, B. Thomson, and S. Young. 2007. Error Simulation for Training Statistical Dialogue Systems. In *Proceedings of ASRU Workshop.*

Y. Tsuruoka, and J. Tsujii. 2005. Bidirectional Inference with the Easiest-First Strategy for Tagging Sequence Data. In *Proceedings of HLT/EMNLP.*

X. Yao, P. Bhutada, K. Georgila. K. Sagae, R. Artstein, and D. Traum. 2010. Practical Evaluation of Speech Recognizers for Virtual Human Dialogue Systems. In *Proceedings of LREC.*